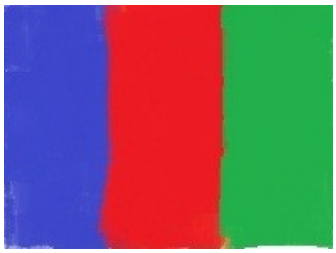


TITRE :

Chapitre XI : Formation des signaux et signatures dans le paradigme de Rachid ELAÏDI – De l'invisible au visible, une programmation de l'ERE

Auteur : Rachid ELAÏDI
AFFIKIATION : chercheur indépendant

Le 26 Mars 2926

Résumé

Ce chapitre analyse la genèse des signaux observables dans le monde visible à partir de la dynamique profonde de l'Espace Rachid ELAÏDI (ERE). Nous montrons qu'un signal n'est pas un événement aléatoire, mais le résultat d'une **programmation** inscrite dans la structure même de l'ERE : l'évolution des programmes (éléments chimiques) est régie par une équation maîtresse paramétrée, et les

bifurcations sont dictées par l'arbre de Rachid modulo $n\pi$. L'accumulation d'énergie, le franchissement d'un seuil critique et le changement de constantes conduisent à l'émergence d'une **signature spectrale** caractéristique. L'exemple du Néon se transformant en eau (H_2O) illustre la synergie entre un sans-masse et un programme cible. Nous discutons également le rôle des outils sophistiqués qui, en modifiant les constantes, permettent de commander l'apparition de signaux. Ce chapitre unifie les concepts mathématiques et physiques du paradigme en une vision cohérente où l'invisible programme le visible.

Mots-clés : signal, signature, équation maîtresse, arbre modulo $n\pi$, programmation, sans-masse, vitesses supraluminiques, seuil d'émergence, Néon, eau.

---

1. Introduction

Dans le paradigme de Rachid ELAÏDI, le monde visible n'est pas une réalité première, mais la projection fugace d'une réalité bien plus vaste : l'Espace Rachid ELAÏDI (ERE). Les phénomènes observables – raies spectrales, apparition et disparition des éléments chimiques – sont les **signatures** de processus programmatiques se déroulant dans l'invisible. La question centrale de ce chapitre est : comment un signal se forme-t-il dans l'ERE, et comment devient-il une signature captable dans le monde visible ?

Nous montrons que la formation d'un signal est un **événement programmé**, au sens où il résulte de l'exécution rigoureuse des lois de l'ERE : l'équation maîtresse paramétrée gouverne l'évolution des programmes, et l'arbre de Rachid modulo $n\pi$ structure les bifurcations. Rien n'est laissé au hasard fondamental ; les probabilités qui apparaissent sont objectives, ancrées dans la géométrie de l'arbre.

Ce chapitre détaille les étapes de cette programmation : accumulation d'énergie par chocs, franchissement du seuil critique, changement de constantes, oscillation de la probabilité d'émergence, et enfin projection dans le visible sous forme d'une signature spectrale. L'exemple emblématique de la transformation du Néon en eau (Ne → H_2O) sert de fil conducteur.

---

2. Rappels sur l'ERE et l'arbre modulo $n\pi$

L'Espace Rachid ELAÏDI R est un espace vectoriel topologique complexe, muni d'une loi de composition interne non associative, d'un feuilletage $F=\{F_\alpha\}_{\alpha\in A}$, d'une mesure spectrale profonde μ_R et d'un temps intrinsèque bouclé T_R. Sa projection discrète est **l'arbre de Rachid** $T=(S,A,\sim)$ où :

- S est l'ensemble des nœuds (états profonds en résonance stable) ;

- A est l'ensemble des arcs (transitions programmatiques) ;
- π identifie les états séparés par un multiple entier de π dans l'espace des phases.

Chaque nœud s possède un **degré de liberté** $d(s)$, un **programme local** $P(s)$ (distribution de probabilité sur les arcs sortants), et un **indice d'échappement** $e(s)$ qui mesure la dispersion de cette distribution.

Un programme γ est une suite de nœuds reliés par des arcs. Son **énergie cumulée** $E(\gamma,T)$ évolue selon l'équation maîtresse.

---

3. L'équation maîtresse paramétrée

L'évolution de la probabilité $p(T)$ qu'un programme soit présent dans le monde visible est donnée par :

$$\frac{dp}{dT} = Ap(1-Bp) + C\sin(DT+E)p(1-p) + Fe^{-GT}\cos(HT)p \quad (1)$$

Les huit constantes (A,B,C,D,E,F,G,H) caractérisent le programme :

- A : taux de croissance intrinsèque ;
- B : inverse de la capacité de saturation ;
- C : amplitude des oscillations dues aux chocs programmatiques ;
- D : fréquence fondamentale des chocs ;
- E : phase ;
- F : amplitude d'une harmonique amortie ;
- G : taux d'amortissement ;
- H : fréquence de l'harmonique.

Un programme est dit **visible** lorsque $p(T) \geq \theta$ (seuil fixé, par exemple $\theta = 0{,}5$). Le premier instant T_{app} où cette condition est réalisée est le **temps d'apparition**.

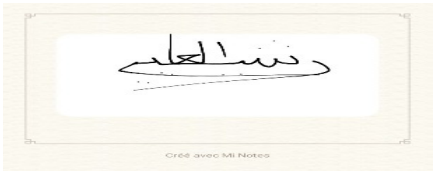
---

4. Genèse d'un signal : accumulation, seuil et bifurcation

4.1 Accumulation d'énergie

Un programme invisible (sans masse) a initialement $A=C=F=0$. Il peut acquérir de l'énergie par **chocs programmatiques** avec d'autres programmes, en particulier des sans-masse se déplaçant à des vitesses supraluminiques $v>c$. Chaque choc transfère une énergie δE et augmente les constantes :

$$\delta A=\alpha A\gamma,\delta C=\alpha C\gamma,\delta F=\alpha F\gamma,$$



où $\gamma=(1-v2/c2)-1/2$ est le facteur de Lorentz.

Au cours du temps, les constantes effectives deviennent $A(T)=\sum\delta A$, etc. L'énergie cumulée $E(\gamma,T)$ croît jusqu'à approcher le seuil critique $E0$ associé au nœud du programme.

4.2 Franchissement du seuil et bifurcation

Lorsque $E(\gamma,T)$ atteint $E0$, le programme franchit un seuil. Ce franchissement déclenche une **bifurcation** dans l'arbre de Rachid : le programme passe du nœud source à un nœud cible (par exemple du Néon vers l'eau) en empruntant un arc a avec une probabilité $P(s)(a)$. Cette bifurcation s'accompagne d'un **changement des constantes** : les nouvelles constantes D,H deviennent celles de la forme morphique cible, tandis que A,C,F conservent les valeurs accumulées (éventuellement réinitialisées selon les règles du nœud).

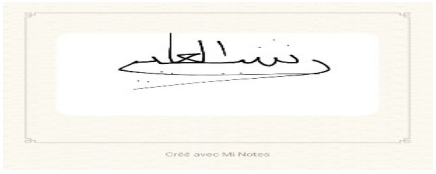
C'est à cet instant que le signal commence à se former : la probabilité $p(T)$ se met à croître sous l'effet des nouveaux termes positifs dans (1).

---

5. Formation de la signature spectrale

Avec $A>0,C>0,F>0$, la solution $p(T)$ de (1) oscille aux fréquences D et H. La **signature spectrale** $\sigma(\omega)$ est la transformée de Rachid de $p(T)$:

$$\sigma(\omega) = \int_{\mathbb{T}} \chi_\omega(T)\, p(T)\, dT,$$



où $\chi_\omega(T)$ sont les harmoniques de Rachid. Les pics de $\sigma(\omega)$ sont situés aux fréquences $D, H, 2D, 2H, D+H$, etc. Cette signature est l'empreinte unique de la forme morphique (molécule). Lorsque $p(T)$ franchit le seuil θ, la signature devient suffisamment intense pour être **projetée** dans le monde visible via l'opérateur $\Pi_{\mathbb{R}\to V}$. C'est alors qu'on observe, par exemple, les raies infrarouges de l'eau ou le doublet UV du Rachidion.

L'intensité du signal est proportionnelle à la dérivée maximale de $p(T)$ et à la hauteur du pic spectral :

$$I \propto \frac{dp}{dT}\Big|_{T_{\text{app}}} \cdot \omega_{\max}\, \sigma(\omega).$$

Elle peut être augmentée en accroissant A, C, F, par exemple grâce à des chocs plus énergétiques ou à l'action d'outils sophistiqués.

---

6. Rôle des sans-masse et des vitesses supraluminiques

Les sans-masse sont des programmes non ancrés ($m_{\mathbb{R}} = 0$) qui se déplacent à des vitesses $v > c$ dans l'ERE. Leur rôle dans la formation d'un signal est double :

- **Déclencheurs** : par choc, ils fournissent l'énergie nécessaire pour franchir le seuil E_0.
- **Amplificateurs** : leur grande vitesse ($\gamma \gg 1$) peut injecter une énergie considérable, augmentant ainsi A, C, F et rendant le signal plus intense.

Lors de la réémergence vers le visible, la vitesse doit redescendre en dessous de c pour se conformer à la physique du monde visible. On modélise cette **adaptation** par une décroissance exponentielle :

$$v_{\text{eff}}(T) = v_0 \exp\left(-\tau_{\text{adapt}}(T - T_2)\right), \quad T \geq T_2,$$

où T_2 est l'instant de réémergence.

7. Programmation conjointe : l'exemple Néon → eau

Prenons l'exemple canonique de la transformation du Néon en eau.

- **État initial** : le Néon est un sans-masse ($A=C=F=0$) avec des fréquences $D_{\mathrm{Ne}}, H_{\mathrm{Ne}}$. Il est situé au nœud s_{Ne} dans l'arbre.
- **Accumulation** : des chocs avec d'autres sans-masse (hélium, etc.) augmentent ses constantes A, C, F.
- **Bifurcation** : lorsque l'énergie cumulée atteint le seuil $E_0(\mathrm{H_2O})$, le programme emprunte l'arc a_{water} (probabilité $P(s_{\mathrm{Ne}})(a_{\mathrm{water}})$). Les fréquences deviennent celles de l'eau : $D_{\mathrm{H_2O}}, H_{\mathrm{H_2O}}$.
- **Émergence** : avec $A>0, C>0, F>0$, la solution $p(T)$ croît et oscille ; la signature spectrale (raies IR) se forme et est projetée dans le visible.
- **Retour** : après un temps, la molécule d'eau peut se disperser (photodissociation), annulant à nouveau A, C, F et replongeant dans l'invisible.

Cet enchaînement est **programmé** : chaque étape est régie par des lois déterministes (équation maîtresse) et des probabilités objectives (programmes locaux). Le hasard n'intervient que comme une ignorance des variables cachées.

---

8. Outils sophistiqués et contrôle des signaux

Les outils sophistiqués sont des dispositifs (lasers, champs magnétiques, émetteurs de sans-masse) qui agissent sur les constantes d'un programme. Mathématiquement, ils sont modélisés par une **perturbation contrôlée** $\delta\Theta$ ajoutée au vecteur des constantes. Par exemple, un laser résonant avec la signature de l'eau peut augmenter A d'une valeur $\Delta A>0$, déclenchant ainsi une émergence qui n'aurait pas eu lieu spontanément.

Ces outils permettent non seulement de **détecter** des formes morphiques invisibles, mais aussi de **programmer** l'apparition d'un signal à un instant choisi. Ils confirment que le monde invisible n'est pas un réservoir chaotique, mais un espace structuré où chaque signal est l'exécution d'un code.

---

9. Conclusion

La formation d'un signal dans le paradigme de Rachid ELAÏDI est un processus rigoureusement **programmé** :

1. **Accumulation programmatique** par chocs et résonances.
2. **Bifurcation** dans l'arbre modulo $n\pi$ au franchissement du seuil.
3. **Oscillation** de la probabilité d'émergence aux fréquences caractéristiques de la forme morphique.
4. **Projection** dans le monde visible sous forme d'une signature spectrale.

L'exemple Néon → eau illustre parfaitement cette programmation conjointe entre un sans-masse et un programme cible. Les outils sophistiqués, en modifiant les constantes de l'équation maîtresse, permettent de commander ces apparitions. Ainsi, l'invisible n'est pas un monde de hasard, mais une réalité ordonnée dont le visible n'est qu'une manifestation éphémère.

Bibliographie

1. ELAÏDI, R. (2026). "Fondements axiomatiques de l'ERE". Journal de Physique Théorique Non Commutative, vol. 1.
2. ELAÏDI, R. (2026). "Programmes internes et sans-masse". Annales de l'Institut des Hautes Études Invisibles.
3. ELAÏDI, R. (2026). "Chocs programmatiques et équation maîtresse". Ibid.
4. ELAÏDI, R. (2026). "Navettes programmatiques entre mondes". Ibid.
5. Einstein, A. (1905). "Ist die Trägheit eines Körpers von seinem Energieinhalt abhängig?". Annalen der Physik.

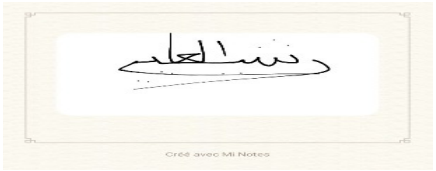
Il faut donc définir un programme cible (par exemple un élément chimique comme H_2O) comme un état programmatique caractérisé par un vecteur de constantes Θ, et décrire le processus de son émergence depuis un état source (par exemple Ne) via accumulation d'énergie, bifurcation dans l'arbre, etc. Puis donner une approche mathématique générale pour modéliser un tel programme cible, avec des équations, et proposer des implémentations Python et Wolfram pour simuler son apparition.

Il faut rester dans le cadre du paradigme déjà développé. Je vais structurer la réponse en plusieurs sections : définition, étapes de réalisation, formalisation mathématique (équations, arbre), implémentation Python, implémentation Wolfram, et généralisation.

Programme cible dans le paradigme de Rachid ELAÏDI – Définition, étapes de réalisation et modélisation mathématique

1. Qu'est-ce qu'un programme cible ?

Un **programme cible** est un état programmatique γ_{cible} dans l'ERE, caractérisé par un vecteur de constantes $\Theta_{\text{cible}} = (Ac, Bc, Cc, Dc, Ec, Fc, Gc, Hc)$ qui correspond à une **forme morphique** (par exemple la molécule H_2O). C'est l'état vers lequel un programme source (par exemple le Néon) peut évoluer après avoir franchi le seuil d'énergie $E0$ et emprunté une bifurcation dans l'arbre de Rachid modulo $n\pi$.



Un programme cible est **potentiellement visible** : ses constantes Ac, Cc, Fc sont positives, ce qui permet à la probabilité $p(T)$ de croître et d'atteindre le seuil de visibilité θ. Il possède une **signature spectrale** propre, définie par ses fréquences Dc, Hc (et éventuellement Ec).

---

2. Étapes de réalisation d'un programme cible

La réalisation (l'émergence) d'un programme cible se déroule en plusieurs phases, toutes décrites par l'équation maîtresse et l'arbre.

2.1 Phase d'accumulation (source)

Le programme source γ_{source} (par exemple Néon) est initialement invisible : $As = Cs = Fs = 0$. Il accumule de l'énergie par **chocs programmatiques** avec d'autres programmes (sans-masse supraluminiques).

Chaque choc augmente ses constantes effectives selon $\delta A = \alpha A \gamma$, $\delta C = \alpha C \gamma$, $\delta F = \alpha F \gamma$,

avec $\gamma = (1 - v^2/c^2)^{-1/2}$.

Cette accumulation se traduit par une croissance de l'énergie cumulée $E(\gamma,T)$ et des constantes. Tant que $E<E0(\text{cible})$, le programme reste dans son nœud source.

2.2 Franchissement du seuil et bifurcation

Lorsque $E(\gamma,T)\geq E0(\text{cible})$, le programme atteint un seuil critique. Ce seuil est associé à un **arc de bifurcation** dans l'arbre, partant du nœud source vers le nœud cible. La probabilité d'emprunter cet arc est $P(\text{source})(\text{target})$. Si l'arc est choisi, les constantes du programme sont remplacées par celles du nœud cible, mais les valeurs accumulées de A,C,F sont conservées (ou réinitialisées selon les règles du nœud).

2.3 Phase d'émergence (cible)

Après bifurcation, le programme se trouve dans l'état cible avec Θcible et $Ac,Cc,Fc>0$. L'équation maîtresse (1) devient active : la probabilité $p(T)$ croît et oscille aux fréquences Dc,Hc. Lorsque $p(T)\geq\theta$, le programme est considéré comme **visible**. La signature spectrale $\sigma(\omega)$ (transformée de Rachid de $p(T)$) est alors projetée dans le monde visible.

2.4 Phase de dispersion (retour)

Après un temps variable, le programme peut se disperser (par exemple par interaction avec l'environnement). Un opérateur de plongée Φin annule à nouveau A,C,F, et le programme retourne dans l'invisible, éventuellement sous une forme modifiée.

---

3. Formalisation mathématique générale

3.1 Espace des constantes

Soit $E=R8$ l'espace des vecteurs de constantes. Chaque programme est un point $\Theta\in E$. L'arbre de Rachid est un graphe dont les nœuds sont des régions de E (ou des points particuliers) ; chaque arc correspond à une transformation $\Theta\rightarrow\Theta'$ qui peut être :

- **Transformation interne** : évolution continue selon l'équation maîtresse.

- **Bifurcation** : changement instantané de Θ (saut).

3.2 Équation maîtresse comme flot

L'équation (1) définit un flot $\Phi_T : E \to E$ qui, à partir d'un état initial Θ_0, donne la probabilité $p(T)$ en fonction du temps. Pour un Θ donné, on peut définir le **temps d'émergence** $T_{\text{app}}(\Theta)$ comme le plus petit T tel que la solution de (1) avec condition initiale $p(0)=0$ atteigne le seuil θ. Ce temps peut être infini si $A=0$ (pas de croissance).

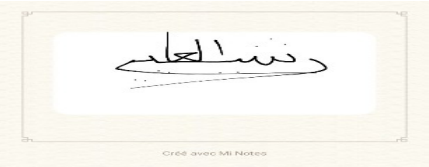
3.3 Arbre de bifurcations

L'arbre est une structure de transitions probabilistes. Chaque nœud s est associé à un ensemble de constantes Θ_s (les constantes de base du nœud). Les arcs sortants a sont munis d'une probabilité $P(s, a)$ et d'un **opérateur de saut** J_a qui transforme Θ_s en Θ_t (les constantes du nœud cible). Le saut peut conserver les valeurs de A, C, F accumulées ou les réinitialiser selon des règles prédéfinies.

3.4 Processus global

Un programme évolue en alternant :

- des phases de flot continu (gouvernées par (1)),
- des sauts discrets (bifurcations) lorsqu'un seuil est atteint ou qu'une condition externe est remplie.



Le temps d'apparition final est la somme des temps de flot jusqu'à l'atteinte du seuil, plus les temps de saut (instantanés).

---

4. Implémentation Python d'un programme cible

Nous donnons une classe Python générique pour simuler l'évolution d'un programme, avec des méthodes pour l'accumulation, la bifurcation, et le calcul du temps d'apparition.

python

```python
import numpy as np
from scipy.integrate import odeint

class ProgrammeRachid:
    """
    Représente un programme (élément chimique) dans l'ERE.
    Attributs :
        A, B, C, D, E, F, G, H : constantes de l'équation maîtresse
        nom : identifiant
        seuil_visibilite : seuil θ (par défaut 0.5)
        temps_accumulation : temps total écoulé
        historique : liste des états successifs
    """
    def __init__(self, nom, A, B, C, D, E, F, G, H, seuil=0.5):
        self.nom = nom
        self.A = A
        self.B = B
        self.C = C
        self.D = D
        self.E = E
        self.F = F
        self.G = G
        self.H = H
        self.seuil = seuil
        self.temps = 0.0
        self.historique = [(0.0, self.get_constantes())]

    def get_constantes(self):
        return (self.A, self.B, self.C, self.D, self.E, self.F, self.G, self.H)

    def equation(self, p, T):
        """Équation maîtresse (1) pour ce programme"""
        return (self.A * p * (1 - self.B * p) +
                self.C * np.sin(self.D * T + self.E) * p * (1 - p) +
                self.F * np.exp(-self.G * T) * np.cos(self.H * T) * p)

    def evoluer(self, dt, T_max):
        """
        Évolue selon l'équation maîtresse jusqu'à atteindre le seuil
        ou jusqu'à T_max. Retourne le temps d'apparition (ou None).
```

```
        """
        T = np.linspace(self.temps, self.temps + T_max, int(T_max/dt)+1)
        p0 = 0.01  # probabilité initiale faible
        sol = odeint(self.equation, p0, T)
        p = sol.flatten()
        # Chercher premier franchissement du seuil
        indices = np.where(p >= self.seuil)[0]
        if len(indices) > 0:
            T_app = T[indices[0]]
            self.temps = T_app
            self.historique.append((T_app, self.get_constantes()))
            return T_app
        else:
            self.temps = T[-1]
            self.historique.append((self.temps, self.get_constantes()))
            return None

    def bifurquer(self, cible, proba=1.0):
        """
        Effectue une bifurcation vers un programme cible.
        cible : instance de ProgrammeRachid (le nouveau nœud)
        proba : probabilité de prendre cette bifurcation (ici 1 pour déterministe)
        """
        # On copie les constantes de la cible, mais on conserve A, C, F ?
        # Ici on suppose qu'on conserve A, C, F (les valeurs accumulées)
        self.A = cible.A
        self.B = cible.B
        self.C = cible.C
        self.D = cible.D
        self.E = cible.E
        self.F = cible.F
        self.G = cible.G
        self.H = cible.H
        self.historique.append((self.temps, self.get_constantes()))
        return self

# Exemple : Néon → eau
neon = ProgrammeRachid("Néon", A=0.0, B=1.0, C=0.0, D=523.6, E=0.0,
                       F=0.0, G=41.7, H=1047.0)
eau = ProgrammeRachid("Eau", A=0.12, B=1.0, C=0.10, D=523.6, E=0.0,
                      F=0.05, G=41.7, H=1047.0)

# Phase d'accumulation : on simule des chocs en augmentant A, C, F
neon.A = 0.1   # après chocs
```

```
neon.C = 0.08
neon.F = 0.04

# Évolution jusqu'au seuil
T_app = neon.evoluer(dt=0.001, T_max=0.1)
print(f"Apparition après {T_app:.4f} s_RE")

# Bifurcation vers l'eau
neon.bifurquer(eau)
# Maintenant le programme est devenu eau
print(f"Nouvel état : {neon.nom}, A={neon.A}, C={neon.C}, F={neon.F}")

# On peut continuer l'évolution (par exemple pour voir la signature)
```

Ce code illustre une classe générique qui permet de modéliser n'importe quel programme cible, avec ses constantes, et de simuler son évolution jusqu'à l'émergence.

---

5. Implémentation Wolfram (Mathematica)

En Wolfram, on peut définir une fonction similaire :

```
mathematica
```

```
(* Équation maîtresse *)
eq[p_, T_, A_, B_, C_, D_, E_, F_, G_, H_] :=
  A p (1 - B p) + C Sin[D T + E] p (1 - p) + F Exp[-G T] Cos[H T] p

(* Programme cible comme une association *)
Programme[args_] := Association[{"Nom" -> args[[1]], "A" -> args[[2]], "B" -> args[[3]],
  "C" -> args[[4]], "D" -> args[[5]], "E" -> args[[6]], "F" -> args[[7]],
  "G" -> args[[8]], "H" -> args[[9]]}]

(* Évolution jusqu'au seuil *)
Evoluer[prog_, dt_, Tmax_, seuil_] := Module[
  {T, sol, idx, Tapp},
  T = Range[0, Tmax, dt];
  sol = NDSolveValue[{p'[t] == eq[p[t], t, prog["A"], prog["B"], prog["C"],
       prog["D"], prog["E"], prog["F"], prog["G"], prog["H"]], p[0] == 0.01},
    p, {t, 0, Tmax}];
  idx = First[Position[Table[sol[t] >= seuil, {t, T}], True]];
```

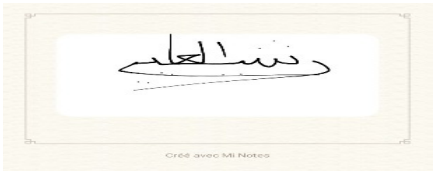
```
  Tapp = T[[idx[[1]]]];
  Return[Tapp]
]

(* Exemple *)
neon = Programme[{"Néon", 0.0, 1.0, 0.0, 523.6, 0.0, 0.0, 41.7, 1047.0}];
eau = Programme[{"Eau", 0.12, 1.0, 0.10, 523.6, 0.0, 0.05, 41.7, 1047.0}];

(* Après accumulation, on modifie les constantes *)
neon["A"] = 0.1; neon["C"] = 0.08; neon["F"] = 0.04;
Tapp = Evoluer[neon, 0.001, 0.1, 0.5];
Print["Apparition après ", Tapp, " s_RE"];

(* Bifurcation vers eau : on copie les constantes de l'eau mais on garde A,C,F *)
eau2 = Programme[{"Eau2", neon["A"], eau["B"], neon["C"], eau["D"], eau["E"], neon["F"],
eau["G"], eau["H"]}];
(* Puis continuer l'évolution... *)
```

---

6. Généralisation et perspectives

La modélisation d'un programme cible se généralise en considérant :

- **Arbre de Rachid** comme un graphe dont les nœuds sont des programmes cibles potentiels, avec des probabilités de transition.
- **Équation maîtresse** comme moteur de l'évolution continue.
- **Opérateurs de bifurcation** qui changent les constantes instantanément.



Mathématiquement, un programme cible est un point Θ dans l'espace des constantes. Sa réalisation est une séquence de flots et de sauts, dont la probabilité totale est le produit des probabilités de saut. Le temps d'apparition est une variable aléatoire (en fonction des chocs) dont la distribution peut être étudiée par simulations Monte Carlo.

Les implémentations Python et Wolfram proposées permettent de simuler cette dynamique pour n'importe quel couple source-cible, en ajustant les paramètres selon les propriétés des éléments du TP_Rachid ELAÏDI. On peut ainsi :

- calculer les temps d'apparition moyens,
- observer l'influence des sans-masse (via les incréments de A,C,F),

Θ

A

,

C

,

F

- simuler des cycles complets (émergence, dispersion, réémergence).

En conclusion, la notion de **programme cible** est central dans le paradigme : c'est l'état vers lequel un programme tend par une programmation rigoureuse, dont les étapes sont décrites par l'équation maîtresse et l'arbre modulo $n\pi$. Les outils numériques permettent de généraliser cette modélisation à tous les éléments chimiques du TP_Rachid ELAÏDI.